"Express Mail" mailing label number:

EV 335895461 US

METHOD AND SYSTEM FOR SCALING IMAGES

Steve Zhihua Zeng

BACKGROUND

[0001] Scaling images is known to accommodate display of an image at an output resolution that is different than the images input resolution. Typically, a large number of filter taps for a single phase filter or a large number of phases for a polyphase filter are needed to accommodate scaling over a wide range of resolutions. Single phase filters usually require much more processing power and memory than polyphase filters. But the large number of phases required for a polyphase filter may still be too expensive for practical implementations. For example, to accommodate downscaling from 720 to lower resolution that is a multiple of 16, without any phase distortion, requires storing coefficients of up to 44 phases, and each of these phases contains a number of filter taps. A method and/or system capable of providing high quality scaling of any scaling ratio with constrained resources would be useful.

Field of the Disclosure

[0002] The present disclosure relates generally to image/video processing and more specifically to scaling of image/video.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] The present disclosure may be better understood, and its advantages made apparent to those skilled in the art by referencing the accompanying drawings.

[0004] FIG. 1 illustrates in flow diagram form a method in accordance with the present disclosure;

[0005] FIG. 2 illustrates a data structure in accordance with the present disclosure;

[0006] FIG. 3. illustrates a flow diagram for a method in accordance with the present disclosure.

[0007] FIG. 4 illustrates a specific embodiment of a method in accordance with the present disclosure; and

[0008] FIG. 5 illustrates a specific embodiment of a system in accordance with the present disclosure.

[0009] The use of the same reference symbols in different drawings indicates similar or identical items.

DESCRIPTION OF THE PREFERRED EMBODIMENT(S)

[0010] A method of scaling images is disclosed in accordance with a specific embodiment of the disclosure. The method provides lossless scaling or scaling having low-level phase distortion for all resolutions.

[0011] FIG. 1 illustrates, in flow diagram form, a method in accordance with a specific embodiment of the present disclosure. At step 51, a number of phases to be made available for scaling is determined. Typically, the number of available phases will be selected deterministically based upon a desired level of quality needed. For example, for purposes of illustration herein, it has been determined that there are to be seventeen (17) available phases. These seventeen phases, or a subset of the seventeen phases, will be used to scale images of any resolution. Because of the symmetric nature of the phases, nine sets of coefficients (Filter Phases) can be provided to support the 17 available phases. This results in an amount of memory needed to support implementing the available phases that is much less than that of the worst scenario that uses 44 phases to scale to any resolution that is a multiple of 16. It will be appreciated that the number of available phases selected can vary depending upon many different factors, such as a desired image quality and the range of resolutions supported.

[0012] At step 52, a control word to be accessed by a scaler is determined. FIG. 2 illustrates a data structure 60 that includes control word (CW) 60 and coefficients for a plurality of filter phases 64. The control word 60 includes a plurality of variables, including: a number of input pixels in a scaling cycle (M), a number of output pixels in a scaling cycle (L), a number of used phases (N), and a shift variable S. The variables associated with control word 60 are better understood with reference to FIG. 3.

[0013] At step 202, of FIG. 3, an input resolution and an output resolution is determined and a greatest common denominator (GCD) of the input and output resolution is determined. Typically the input resolution is determined by accessing a register location or receiving the resolution as part of a video stream, or video file. The output resolution is typically determined by accessing a register, or receiving the value from a display driver. Assuming an input resolution of 720 pixels, and an output resolution of 704 pixels, there is a GCD of 16. For purposes of discussion, the horizontal dimension of an image is discussed herein

[0014] At step 204, the variable M is set to the value derived from dividing the input resolution by the GCD. Based upon the example illustrated, M = (720 / 16), or 45. The value 45 represents the number of input pixels that are scaled during a scaling cycle. At step 206, the variable L is set to the value derived by dividing the output resolution by the GCD. Based upon the example illustrated, L = (704 / 16), or 44. The value 44 represents the number of pixels generated during a scaling cycle. A scaling cycle represents a scaling operation that is repeated, such that each cycle accesses a common set of filter phases to scale M input pixels to generate L output pixels. In the illustrated example, each scaling cycle will result in 45 input pixels being scaled to 44 output pixels. In order to scale an entire line of 720 pixels, 16 scaling cycles are implements.

[0015] Steps 208, 210, and 212 determine a value of the shift variable S. At step 208 S is initialized to zero. At step 210, a determination is made whether a result of right-shifting the binary representation of the number of output pixels (L) by the value of S results in a value that is less than the number of available phases, or seventeen (17) in the present example. For example, when S is equal to zero, the binary value of the number of input pixels is, in effect, not right-shifted, resulting in a decimal value of 44. Since 44 is not less than the number of available phases, seventeen, the flow proceeds to step 212, where the shift value S is incremented by one.

[0016] Flow proceeds from step 212 to step 210, where a determination is made whether the binary representation of L right-shifted by the current value of S, now one, is less than the number of available phases. For example, during the current loop, when S is equal to one, the binary value of the number of output pixels in one scaling cycle (0010 1100) is shifted by one, resulting in a value of '0001 0110' or decimal 22. Since 22 is not less than the number of available phases, seventeen, the flow proceeds again to step 212, where the shift value S is incremented by one. This next loop, with S equal to two, results in a shifted binary value of '0000 1011', or decimal 11. Since 11 is less than the number of available phases, decimal seventeen, the value of S is determined and flow proceeds to step 214.

[0017] At step 214, the number of used phases (N) is determined by right-shifting the binary representation of L by the value of S.

[0018] It will be appreciated that in one embodiment each of the values of M, L, N, and S are provided as part of the control word 60. However, it will also be appreciated that only two of the variables L, N and S need to be provided as part of the control word N. By providing only two variables, the third, unprovided, variable can be derived using the equation N = L >> S, where ">>" indicates a number of right-shifts (S) to apply to the binary representation of L. Possible control word variables, therefore, include providing any one of the variable sets MLNS, MLN, MLS, and MNS.

[0019] Once the control word variables are known, whether provided, or provided and calculated, as described above, the method described in FIG. 4 can be used to scale an image. The method of FIG. 4 is discussed with references to scaling in a horizontal dimension by implementing one or more scaling cycles. For example, sixteen scaling cycles are used to scale 720 pixels to 704 pixels, each scaling cycle generating 44 output pixels. Each scaling cycle accesses the FILTER PHASES 64 in a common sequence.

[0020] Initially, at step 310, a current phase variable, labeled CURRENT_PHASE, is set to zero. At step 212, a phase index variable, labeled PHASE_INDEX, is set equal to the binary representation of CURRENT_PHASE right shifted by value of S. The value of PHASE_INDEX is less than or equal to the number of available phases, and is used to access a specific set of phase coefficients from the data structure 60 to be used during the current phase scaling.

[0021] At step 314, a determination is made whether the PHASE_INDEX is greater than the number of used phases (N) divided by two (N / 2). If so, the flow proceeds to step 332, otherwise the flow proceeds to step 316.

[0022] When the PHASE_INDEX is greater than N / 2, the set of coefficients to be used for the current phase scaling needs to be accessed from a mirrored location. The value of PHASE_INDEX is mirrored at step 332 by subtracting PHASE_INDEX from the number of used phases (N). For example, for a used number of phases equal to 11, a PHASE_INDEX of eight would be mirrored resulting in a PHASE_INDEX equal to three (11 – 8 = 3). Referring to FIG. 2, this would result in the coefficients of FILTER PHASE 3 being accessed during the scaling of a current pixel.

[0023] Once the mirrored location is accessed, the flow proceeds to step 314, where the coefficients accessed from the mirrored location are reversed prior to use.

[0024] When the PHASE_INDEX is not greater than N / 2, the set of coefficients to be used for scaling is accessed from a direct location at step 316. For example, given a used number of phases equal to 11, a PHASE_INDEX of 4 would result in a direct access of the coefficients of FILTER PHASE 4, as opposed to an access from a mirror location.

[0025] At step 318, a filter with the selected phase is used to filter one pixel, resulting in a scaled pixel. The specific implementation illustrated in FIG. 2 provides eight coefficients with each FILTER PHASE 64 for use by an eight-tap filter that performs the scaling. The eight coefficients illustrated in FIG.2 include two eight-bit signed coefficients, two nine-bit signed coefficients, two ten-bit signed coefficients, one ten-bit unsigned coefficient, and one eleven-bit unsigned coefficient. As previously discussed, when a phase is mapped to its symmetric (mirrored) position, the coefficients at the mirror location need to be reversed from left to right. The left center is an eleven-bit unsigned coefficient to handle the no scaling case. More specifically, the normalized value of all the coefficients (the sum of all the coefficients) is eleven-bit in this specific design. When there is no scaling, normalized value is placed into the left center.

[0026] At step 120, a determination is made whether an end of line has been reached. If so, the flow proceeds to step 310 and scaling of a new line begins.

[0027] Steps 322, 324, and 326 determine a next value for the CURRENT_PHASE variable. At step 322 the CURRENT_PHASE is incremented by the number of input pixels M. At step 324 a determination is made whether CURRENT_PHASE is greater than the number of output pixels L. If not greater than the variable L, the flow proceeds to step 312, and the previously discussed steps are repeated. If CURRENT_PHASE is greater than the variable L the flow proceeds to step 326 where CURRENT_PHASE is decremented by the variable L, and the input is shifted by one pixel. Once decremented, the flow proceeds to step 324. Once a CURRENT_PHASE value that is not greater than L is achieved, the flow proceeds to step 312, and a new set of coefficients is selected in the manner previously discussed.

[0028] The method of FIG. 4 has been found to be an efficient scaling implementation. Different control words and filter phases can be provided or accessed to implement scaling of different quality levels. While the specific example discussed herein dealt with horizontal scaling, similar techniques can be used to implement vertical scaling. In other embodiments, different scaling techniques can be used with respect to vertical scaling.

[0029] Figure 5 illustrates, in block diagram form, a data processing system that may represent a general purpose processing system, such as a personal computer or a personal digital assistant, or an application specific system such as a media server, internet appliance, home networking hubs, and the like. The system 500 is illustrated to include a central processing unit 510, which may be a conventional or proprietary data processor, memory including random access memory 512, read only memory 514, input output adapter 522, a user interface adapter 520, a communications interface adapter 524, and a multimedia controller 526.

[0030] The input output (I/O) adapter 526 can be further connected to various peripherals such as disk drives 547, printer 545, removable storage devices 546, as well as other standard and proprietary I/O devices.

[0031] The user interface adapter 520 can be considered to be a specialized I/O adapter. The adapter 520 is illustrated to be connected to a mouse 540, and a keyboard 541. In addition, the user interface adapter 520 may be connected to other devices capable of providing various types of user control, such as touch screen devices.

[0032] The communications interface adapter 524 is connected to a bridge 550 such as is associated with a local or a wide area network, and a modem 551. By connecting the system bus 502 to various communication devices, external access to information can be obtained.

[0033] The multimedia controller 526 will generally include a video graphics controller capable of generating scaled images that can be displayed, saved, or transmitted. The specific embodiment illustrated illustrates the multimedia controller including a scaler 570 that can be used to implement the methods described herein. It will be appreciated the methods described can be implemented, by the scaler in hardware and/or software. Software implementations can be stored in any on of various memory locations, including RAM 512 and ROM 514, in addition software implementation software can be stored in the multimedia controller 526. When implemented in software, the scaler may be a data processor within the controller 526 for executing instruction, or it maybe a shared processor, such as CPU 510.

[0034] The elements associated with data structure 60, as described with reference to FIG. 2, can be provided via a storage media, such as floppy 546, Disk Drives 547, or from remote storage devices. The output pixels generated by scaler 570 can be provided to Monitor 560, stored for later display or subsequent processing, or transmitted to a different system. In another embodiment, the scaler 570 can be part of an encoder that generates scaled images for transmission. The input video can be received at a receiver that is part of Multimedia Controller 526, through the Bridge 550 or modem 551, or accessed from a stored memory location. Control words, and coefficients and input pixels can be accessed from system memory locations.

[0035] The preceding detailed description of the figures, reference has been made to the accompanying drawings which form a part thereof, and to which show by way of illustration specific embodiments in which the invention may be practiced. It will be appreciated that many other varied embodiments that incorporate the teachings herein may be easily constructed by those skilled in the art. Accordingly, the present disclosure is not intended to be limited to the specific form set forth herein, but on the contrary, it is intended to cover such alternatives, modifications, and equivalents, as can be reasonably included within the spirit and scope of the invention. The preceding detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present disclosure is defined only by the appended claims.